

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re U.S. Patent Application of )  
YAMAMOTO et al. )  
Application Number: To be Assigned )  
Filed: Concurrently Herewith )  
For: PARALLEL FAST FOURIER TRANSFORMATION )  
METHOD OF CONCEALED-COMMUNICATION TYPE )  
ATTORNEY DOCKET NO. NITT.0193 )

Honorable Assistant Commissioner  
for Patents  
Washington, D.C. 20231

**REQUEST FOR PRIORITY  
UNDER 35 U.S.C. § 119  
AND THE INTERNATIONAL CONVENTION**

Sir:

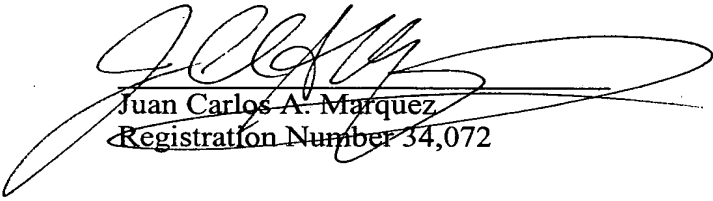
In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of May 23, 2003, the filing date of the corresponding Japanese patent application 2003-145607.

A certified copy of Japanese patent application 2003-145607 is being submitted herewith. Acknowledgment of receipt of the certified copy is respectfully requested in due course.

Respectfully submitted,

\_\_\_\_\_  
Stanley P. Fisher  
Registration Number 24,344

**REED SMITH LLP**  
3110 Fairview Park Drive  
Suite 1400  
Falls Church, Virginia 22042  
(703) 641-4200  
February 25, 2004

  
\_\_\_\_\_  
Juan Carlos A. Marquez  
Registration Number 34,072

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 3 年   5 月 2 3 日  
Date of Application:

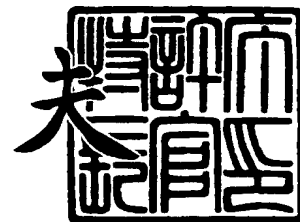
出 願 番 号            特 願 2 0 0 3 - 1 4 5 6 0 7  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 3 - 1 4 5 6 0 7 ]

出   願   人            株 式 会 社 日 立 製 作 所  
Applicant(s):

2 0 0 4 年   2 月 1 3 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号   出証特 2 0 0 4 - 3 0 0 9 1 3 3

【書類名】 特許願

【整理番号】 H03000051A

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/14

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

【氏名】 山本 有作

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

【氏名】 直野 健

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【電話番号】 03-3212-1111

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 通信隠蔽型の並列高速フーリエ変換方法

【特許請求の範囲】

【請求項 1】

入力装置と、それぞれがメモリを備えた複数のプロセッサ及びそれらのメモリ間でデータの転送を行うネットワークからなる処理装置と、出力装置と、外部記憶装置とを含む並列計算機上で高速フーリエ変換を行う方法であって、

変換対象となるデータを複数の部分データに分割し、各部分データを前記複数のプロセッサのメモリに分散して格納するステップと、

各プロセッサで前記複数の部分データをさらに分割した第 1 部分、第 2 部分のうちの各第 1 部分に対して演算処理を行うステップと、

各プロセッサで前記複数の部分データの各第 2 部分に対して演算処理を行い、これと並行して前記各第 1 部分の演算処理結果を前記複数のプロセッサ間で再配置するステップを含むことを特徴とする並列計算機上での高速フーリエ変換方法。

【請求項 2】

前記複数の部分データの各第 1 部分に対する演算処理、及び各第 2 部分に対する演算処理は、いずれもデータを第 1 軸方向に関してフーリエ変換する処理であり、

前記各第 1 部分の演算処理結果の再配分のステップに引き続き、各プロセッサに再配分されたデータを第 2 軸方向に関してフーリエ変換するステップを更に有する請求項 1 の高速フーリエ変換方法。

【請求項 3】

前記複数の部分データの各第 1 部分とはそれぞれ前記変換対象のデータ配列のうち第 2 軸方向で偶数番目のデータであり、各第 2 部分とはそれぞれ前記変換対象のデータ配列のうち第 2 軸方向で奇数番目のデータであることを特徴とする請求項 2 の高速フーリエ変換方法。

【請求項 4】

前記再配分されたデータを第 2 軸方向に関してフーリエ変換するステップに並

行して、前記複数の部分データの各第2部分の第1の方向に関するフーリエ変換の結果を各プロセッサに再配分することを特徴とする請求項3の高速フーリエ変換方法。

【請求項5】

前記各第2部分の第1軸方向に関するフーリエ変換の結果の再配分のステップの終了後に、再配分された各第2部分のデータを第2軸方向に関してフーリエ変換するステップと、

該各第2部分の第2軸方向に関するフーリエ変換の結果と、先に得られた各第1部分の第2軸方向に関するフーリエ変換の結果との双方を用いて、前記変換対象のデータの第2軸方向に関するフーリエ変換の最終演算を行うステップとを更に有する請求項4の高速フーリエ変換方法。

【請求項6】

入力装置と、それぞれがメモリを備えた複数のプロセッサ及びそれらのメモリ間でデータの転送を行うネットワークからなる処理装置と、出力装置と、外部記憶装置とを含む並列計算機上で3次元の高速フーリエ変換を行う方法であって、

データをX軸方向、Y軸方向、Z軸方向の長さがそれぞれNX、NY、NZであるような直方体として並べ、Z方向に垂直な面で分割して各部分データを前記複数のプロセッサのメモリに分散して格納し、

各プロセッサにてX座標が偶数の要素のみに対し、Y方向の変換処理を行い、

各プロセッサにてX座標が奇数の要素のみに対し、Y方向の変換処理を行い、これと並行して、X座標が偶数の要素に対しては、データがY軸に垂直な面で分割されて各部分データが前記複数のプロセッサのメモリに分散して格納されるよう、データの転送処理を行い、

各プロセッサにてX座標が偶数の要素のみに対し、X方向の変換の最初の $\log_2 N$  X - 1 ステップを行い、これと並行してX座標が偶数の要素に対しては、データがY軸に垂直な面で分割されて各部分データが前記複数のプロセッサのメモリに分散して格納されるよう、データの転送処理を行い、

各プロセッサにてX座標が奇数の要素のみに対し、X方向の変換の最初の $\log_2 N$  X - 1 ステップを行い、

各プロセッサにてX方向の変換の最後の1ステップを行い、  
各プロセッサにてZ方向の変換処理を行うことを特徴とする並列計算機上での  
高速フーリエ変換方法。

【請求項7】

入力装置と、それぞれがメモリを備えた複数のプロセッサ及びそれらのメモリ  
間でデータの転送を行うネットワークからなる処理装置と、出力装置と、外部記  
憶装置とから構成される並列計算機上でN点のデータに対する1次元の高速フー  
リエ変換を行う方法であって、

$N_X$ 、 $N_Y$ 、 $N_Z$ を $N_X \times N_Y \times N_Z = N$ を満たす整数とするとき、変換対象のデータをX軸  
方向、Y軸方向、Z軸方向の長さがそれぞれ $N_X$ 、 $N_Y$ 、 $N_Z$ であるような直方体として  
並べ、Z方向に垂直な面でデータを分割し、各部分データを前記複数のプロセッ  
サのメモリに分散して格納し、

各プロセッサにてX座標が偶数の要素のみに対し、Y方向の変換処理とひねり係  
数の乗算処理を行い、

各プロセッサにてX座標が奇数の要素のみに対し、Y方向の変換処理とひねり係  
数の乗算処理を行い、これと並行してX座標が偶数の要素に対しては、データがY  
軸に垂直な面で分割されて各部分データが該並列計算機の各プロセッサのメモリ  
に分散して格納されるよう、データの転送処理を行い、

各プロセッサにてX座標が偶数の要素のみに対し、X方向の変換の最初の $\log_2 N$   
X-1ステップを行い、これと並行してX座標が偶数の要素に対しては、データが  
Y軸に垂直な面で分割されて各部分データが該並列計算機の各プロセッサのメモ  
リに分散して格納されるよう、データの転送処理を行い、

各プロセッサにてX座標が奇数の要素のみに対し、X方向の変換の最初の $\log_2 N$   
X-1ステップを行い、

各プロセッサにおいてX方向の変換の最後の1ステップとひねり係数の乗算処  
理を行い、

各プロセッサにおいてZ方向の変換処理を行うことを特徴とする並列計算機上  
での高速フーリエ変換方法。

【請求項8】

入力装置と、それぞれがメモリを備えた複数のプロセッサ及びそれらのメモリ間でデータの転送を行うネットワークからなる処理装置と、出力装置と、外部記憶装置とを含む並列計算機上で高速フーリエ変換を行う方法であって、

該並列計算機の各プロセッサの演算の状態及びプロセッサ間の通信の状態を表示するモニタで観察した場合に、処理が4つのフェーズから構成され、

第1のフェーズでは各プロセッサは演算のみを行い、

第2のフェーズでは各プロセッサは演算を行うと同時に、自分以外の全てのプロセッサに対してデータの転送処理を行い、

第3のフェーズでは各プロセッサは演算を行うと同時に、自分以外の全てのプロセッサに対してデータの転送処理を行い、

第4のフェーズでは各プロセッサは演算のみを行うことを特徴とする並列計算機上での高速フーリエ変換方法。

#### 【請求項 9】

並列計算機を用いて高速フーリエ変換を利用して気象予測計算を行うシステムであって、該高速フーリエ変換の部分に請求項1、請求項6、請求項7もしくは請求項8のいずれか一つに記載の方法を使うことを特徴とする並列計算機上での気象予測計算システム。

#### 【請求項 10】

並列計算機を用いて高速フーリエ変換を利用して電子構造計算を行うシステムであって、該高速フーリエ変換の部分に請求項1、請求項6、請求項7もしくは請求項8のいずれか一つに記載の方法を使うことを特徴とする並列計算機上での電子構造計算システム。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

本発明は分散メモリ型の並列計算機を用いて高速フーリエ変換を行うための方式に関する。

##### 【0002】

#### 【従来の技術】

半導体デバイスの特性計算、電子状態計算、気象予測のための計算などの科学技術計算では、数万から数百万に上る変数を扱う大規模シミュレーションが必要である。このような大規模問題を扱う手段としては、並列計算機、特に分散メモリ型と呼ばれるタイプの並列計算機が有力である。分散メモリ型並列計算機は、各々がメモリを持つプロセッサをネットワークで複数個結合したシステムであり、従来の逐次型計算機に比べ、プロセッサ台数を増やすことでピーク性能をいくらかでも高めることができるという利点を持つ。

分散メモリ型並列計算機では、計算対象のデータを各プロセッサの持つメモリに分散させて格納し、そのそれぞれに対して各プロセッサが並列に計算を行う。もし計算途中で他のプロセッサの持つデータが必要となる場合は、必要なデータをそのプロセッサから転送してもらい、その後に計算を続行する。したがって分散メモリ型並列計算機では、一般に、演算時間に加えてプロセッサ間でデータを転送するための時間が必要である。そこで、計算効率を上げるには、プロセッサ間での通信がなるべく少なく済むような並列性の高い計算手法が必要である。また、多くの分散メモリ型並列計算機においては、あるデータに対して演算を行っている間に、別のデータを他プロセッサに転送する機構が用意されている。この場合は、演算とデータ転送が同時に行えるように計算手法を工夫すれば、転送のための時間が演算時間に隠蔽でき、計算効率を向上させることができる。

### 【0003】

科学技術計算において頻繁に利用される処理の一つに、フーリエ変換がある。フーリエ変換は、ある実数区間で定義された複素数値をとる関数 $f(x)$ を複素指数関数 $\exp(ikx)$ の重ね合わせとして表す処理であり、計算機上で実現する場合には、扱いうる点の数が有限であることから、複素数の点列 $f_0, f_1, \dots, f_{N-1}$ を $N$ 個の複素指数関数

$\exp(2\pi ikj/N)$  (ただし $k=0, 1, \dots, N-1$ で、 $i$ は虚数単位、 $\pi$ は円周率)

の重ね合わせとして



$$f_j = \sum_{k=0}^{N-1} c_k \exp(2\pi i k j / N) \quad (\text{ただし } j=0, 1, \dots, N-1)$$

と表す処理となる。すなわち、 $f_0, f_1, \dots, f_{N-1}$ が与えられたときに、重ね合わせの係数 $c_0, c_1, \dots, c_{N-1}$ を求める処理がフーリエ変換である。この係数 $c_0, c_1, \dots, c_{N-1}$ は、式

$$c_k = (1/N) \sum_{j=0}^{N-1} f_j \exp(-2\pi i k j / N) \quad (\text{ただし } k=0, 1, \dots, N-1)$$

により計算できることが知られているが、この定義に基づいて計算を行うと、式の数が $N$ 本あり、各式が $N$ 個の項から成るため、複素指数関数  $\exp(-2\pi i k j / N)$  の計算に加えて、複素数の加算と乗算が約 $N^2$ 回必要である。そこで実際には、アルゴリズム上の工夫により計算量を約 $N \log N$ のオーダーに減少させた高速フーリエ変換という手法が広く使われている。高速フーリエ変換の計算法については、たとえばG. Golub and C. F. van Loan: "Matrix Computations", 3rd edition, The Johns Hopkins University Press, 1996, pp.189~192 (非特許文献1) に詳しく記述されている。

以上で述べたフーリエ変換は1次元のフーリエ変換と呼ばれるが、半導体デバイスの特性計算、電子状態計算、気象予測のための計算などでは、3次元のフーリエ変換が用いられる。これは、3つの添字 $j_x, j_y, j_z$ を持つ複素数データ  $\{f_{j_x, j_y, j_z}\}$  (ただし $j_x=0, 1, \dots, N_x-1, j_y=0, 1, \dots, N_y-1, j_z=0, 1, \dots, N_z-1$ ) を $N_x \times N_y \times N_z$ 個の複素指数関数

$$\exp(2\pi i k_x j_x / N_x) \exp(2\pi i k_y j_y / N_y) \exp(2\pi i k_z j_z / N_z) \\ (\text{ただし } k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; k_z=0, 1, \dots, N_z-1)$$

の重ね合わせとして

$$f_{j_x, j_y, j_z} = \sum_{k_x=0}^{N_x-1} \sum_{k_y=0}^{N_y-1} \sum_{k_z=0}^{N_z-1} c_{k_x, k_y, k_z} \exp(2\pi i k_x j_x / N_x) \exp(2\pi i k_y j_y / N_y) \exp(2\pi i k_z j_z / N_z)$$

(ただし  $j_x=0, 1, \dots, N_x-1; j_y=0, 1, \dots, N_y-1; j_z=0, 1, \dots, N_z-1$ )

と表す処理となる。すなわち、 $\{f_{j_x, j_y, j_z}\}$  が与えられたときに、重ね合わせの係数  $\{c_{k_x, k_y, k_z}\}$  を求める処理が 3 次元フーリエ変換である。この係数  $\{c_{k_x, k_y, k_z}\}$  は、次の式によって計算できることが知られている。

$$c_{k_x, k_y, k_z} = \sum_{j_x=0}^{N_x-1} \sum_{j_y=0}^{N_y-1} \sum_{j_z=0}^{N_z-1} f_{j_x, j_y, j_z} \exp(-2\pi i k_x j_x / N_x) \exp(-2\pi i k_y j_y / N_y) \exp(-2\pi i k_z j_z / N_z)$$

(ただし  $k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; k_z=0, 1, \dots, N_z-1$ )

さらに、この式は次の 3 つの変換を順に行うことによって計算できることが容易に示せる。

<Y方向の変換>

$$c_{j_x, k_y, j_z}^{(1)} = \sum_{j_y=0}^{N_y-1} f_{j_x, j_y, j_z} \exp(-2\pi i k_y j_y / N_y)$$

(ただし  $j_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; j_z=0, 1, \dots, N_z-1$ )

<X方向の変換>

$$c_{k_x, k_y, j_z}^{(2)} = \sum_{j_x=0}^{N_x-1} c_{j_x, k_y, j_z}^{(1)} \exp(-2\pi i k_x j_x / N_x)$$

(ただし  $k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; j_z=0, 1, \dots, N_z-1$ )

<Z方向の変換>

$$c_{k_x, k_y, k_z} = \sum_{j_z=0}^{N_z-1} c_{k_x, k_y, j_z}^{(2)} \exp(-2\pi i k_z j_z / N_z)$$

(ただし  $k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; k_z=0, 1, \dots, N_z-1$ )

これらの式から明らかなように、Y方向の変換では、添字  $j_x, j_z$  が同じ  $N_y$  個のデータに対し、1次元のフーリエ変換を行う。そして、 $j_x$  と  $j_z$  とを動かしてこのような変換を  $N_x \times N_z$  回行うことにより、Y方向の変換が完了する。X方向、Z方向の変換についても同様である。したがって、図2の1に示すように、3次元のデータ  $\{f_{j_x, j_y, j_z}\}$  を各辺の長さが  $N_x \times N_y \times N_z$  の直方体状に並べた場合、Y方向の変

換ではY軸に平行な $N_y$ 個のデータ2に対して1次元のフーリエ変換を行い、X方向の変換ではX軸に平行な $N_x$ 個のデータ3、Z方向の変換ではZ軸に平行な $N_z$ 個のデータ4に対してそれぞれ1次元のフーリエ変換を行うことになる。この計算方式より、Y方向の変換においてはX座標あるいはZ座標が異なる組に対する計算は並列に行えることが明らかである。また、X方向の変換においてはY座標あるいはZ座標が異なる組に対する計算が並列に行えること、Z方向の変換においてはX座標あるいはY座標が異なる組に対する計算が並列に行えることも明らかである。

#### 【 0 0 0 4 】

従来、3次元の高速フーリエ変換を分散メモリ型並列計算機で行うに当たっては、この並列性を利用する方式が一般的であった。その中でも最もプロセッサ間でのデータ転送量が少なく効率的な方式は転置アルゴリズムと呼ばれ、たとえば V. Kumar, A. Grama, A. Gupta and G. Karypis: "Introduction to Parallel Computing", The Benjamin/Cummings Publishing Company, 1994, pp.377~406 (非特許文献2) に詳しく解説されている。この方式では、図3に示すように、まず3次元のデータをZ軸に垂直な平面でプロセッサ台数に等しい数の部分データ5に分割し、このデータ5をそれぞれ1台のプロセッサのメモリに格納する。そして、この状態でY方向の変換を行う。この状態では、Y方向の変換に必要な $N_y$ 個のデータ2は1台のプロセッサが持っているため、プロセッサ間でデータ転送を行うことなく変換が実行できることが明らかである。Y方向の変換の終了後、データの分割方式を変更し、3次元のデータをY軸に垂直な平面でプロセッサ台数に等しい数の部分データ6に分割し、このデータ6をそれぞれ1台のプロセッサのメモリに格納する。このためには、全プロセッサが自分以外の全プロセッサとデータの転送を行う処理が必要である。この処理を転置と呼ぶ。転置の終了後には、各プロセッサはX方向の変換のために必要な $N_x$ 個のデータ3をすべて自分で持っているため、プロセッサ間でデータ転送を行うことなくX方向の変換を行うことができる。また、Z方向の変換においても、各プロセッサは変換のために必要な $N_z$ 個のデータ4をすべて自分で持っているため、プロセッサ間でデータ転送を行うことなく実行でき、これによって3次元のフーリエ変換が終了する。以上が分散メモリ型並列計算機による従来の3次元高速フーリエ変換の実行方法である。

## 【非特許文献 1】

G. Golub and C. F. van Loan: "Matrix Computations", 3<sup>rd</sup> edition, The Johns Hopkins University Press, 1996, pp.189~192

## 【非特許文献 2】

V. Kumar, A. Grama, A. Gupta and G. Karypis: "Introduction to Parallel Computing", The Benjamin/Cummings Publishing Company, 1994, pp.377~406

## 【0005】

## 【発明が解決しようとする課題】

上記の転置アルゴリズムによる並列計算方法では、Y方向、X方向、Z方向の各変換が各プロセッサで完全に独立に実行できる。しかし、途中の転置処理においては、全プロセッサ対全プロセッサのデータ転送が必要となる。一般に、分散メモリ型並列計算機では、データ転送は演算に比べて多くの時間を必要とし、この傾向は近年のプロセッサの高速化に伴ってますます強まりつつある。また最近では、分散メモリ型並列計算機的一种として、多数のパーソナルコンピュータ（PC）をイーサネット（登録商標）などのネットワークで結合したPCクラスタが広く使われるようになってきている。この場合は、専用の並列計算機に比べてプロセッサ間でのデータ転送能力が低いため、特にデータ転送処理が処理時間のネックとなりやすい。このような背景から、分散メモリ型並列計算機で3次元の高速フーリエ変換を行う場合、従来の転置アルゴリズムによる方式では十分な並列性能が得られない場合が多数生じてきている。本発明では、この問題を解決することを目的とする。

## 【0006】

## 【課題を解決するための手段】

上記目的を達成するため、本発明では、転置アルゴリズムにおけるY方向とX方向の変換をそれぞれ2つの部分処理に分割して演算の順序を変更し、通信と演算のオーバーラップを可能にする。そのため、変換に関する次の性質に着目する。

- ① Y方向の変換では、X方向に関して並列性がある。したがって、各プロセッサは、自分の行うべき $N_X$ 組の変換を、任意の順序で行うことができる。
- ② X方向の変換は $\log_2 N_X$ 個のステップからなるが、このうちで、最初の $\log_2 N_X$

-1個のステップでは、X座標が偶数の要素は偶数の要素同士、奇数の要素は奇数の要素同士でしか演算を行わない。

$N_X=8$ の場合を例に取り、X方向の変換における1組のデータに対する計算の流れを図4に示す。ここで、縦に並ぶ8個の丸7は各ステップでのデータの1要素を表し、丸と丸を結ぶ線8は、右側の丸の値を計算するために左側の丸の値を使うことを示す。図より明らかなように、変換のうち最初の $\log_2 N_X=2$ 個のステップでは、X座標が偶数の要素は偶数の要素同士、奇数の要素は奇数の要素同士でしか演算を行わない。したがって、上記で述べた性質②が成り立っていることがわかる。この性質は $N_X=8$ の場合に限らず、一般の場合について成り立つ。

上記①、②の2つの性質を利用すると、転置アルゴリズムにおけるY方向と方向の変換を次の5つの処理に分割して行うことができる。

<処理1>

X座標が偶数の要素のみに対し、Y方向の変換を行う。

<処理2>

X座標が奇数の要素のみに対し、Y方向の変換を行う。

X座標が偶数の要素に対しては、転置処理を行う。

<処理3>

X座標が偶数の要素のみに対し、X方向の変換の最初の $\log_2 N_X=1$ ステップを行う。

X座標が奇数の要素に対しては、転置処理を行う。

<処理4>

X座標が奇数の要素のみに対し、X方向の変換の最初の $\log_2 N_X=1$ ステップを行う。

<処理5>

X方向の変換の最後の1ステップを行う。

本発明による処理1～処理4の様子を図5に示す。図において、斜線部分9および10は変換の計算を行っている要素、灰色の部分11はデータ転送を行っている要素、白の部分12は何も行わない要素を示す。また、直方体中の線はプロセッサへのデータ分割を示す。本発明の計算方式では、処理2においてX座標が奇数の要素に対して演算処理を行っている間に、X座標が偶数の要素に対して転置処理を行うことができる。また、処理3においては、X座標が偶数の要素に対して演算

処理を行っている間に、X座標が奇数の要素に対して転置処理を行うことができる。これにより、転送のための時間を演算時間に隠蔽することが可能となり、計算効率を向上させることができる。

【0007】

【発明の実施の形態】

《実施例 1》

(1) 装置の概略構成

以下、本発明の原理および実施例を、図面により詳細に説明する。ここで実施例として挙げるのは、分散メモリ型並列計算機上で、3次元のフーリエ変換を用いて偏微分方程式を解き、これを用いてシミュレーションを行う場合である。ここでは特に、シミュレーションの例として気象予測のための計算を取り上げる。

本方法における並列プログラムを実行する並列計算機システムを図6に示す。システムは計算領域の形状、初期条件、物質定数などのパラメータを入力するための入力装置13、それぞれがメモリ14を備えたP台のプロセッサ15と、各プロセッサの持つメモリの間でデータ転送を行うためのネットワーク16から成る処理装置17、計算結果を出力するための出力装置18、プログラム19およびデータ20を格納するための外部記憶装置21から構成される。また、異なるプロセッサの持つメモリの間でのネットワークを介したデータ転送は、プロセッサの演算処理と同時に実行可能であるとする。

(2) 従来の方法による3次元並列高速フーリエ変換

以下、分散メモリ型並列計算機上での3次元高速フーリエ変換の原理を、数式を用いて説明する。3次元フーリエ変換は、 $N_x \times N_y \times N_z$ 個の複素入力データ  $\{f_{j_x, j_y, j_z}\}$  から  $N_x \times N_y \times N_z$ 個の複素出力データ  $\{c_{k_x, k_y, k_z}\}$  を、(数1)を用いて計算する処理である。

$$c_{k_x, k_y, k_z} = \sum_{j_x=0}^{N_x-1} \sum_{j_y=0}^{N_y-1} \sum_{j_z=0}^{N_z-1} f_{j_x, j_y, j_z} \exp(-2\pi i k_x j_x / N_x) \exp(-2\pi i k_y j_y / N_y) \exp(-2\pi i k_z j_z / N_z) \\ (\text{ただし } k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; k_z=0, 1, \dots, N_z-1) \\ \text{--- (数1)}$$

容易にわかるように、この式は次の(数2)(数3)(数4)でそれぞれ示すY方向の変換、X方向の変換、Z方向の変換を順に行うことによって計算することができる。

$$c_{j_x, k_y, j_z}^{(1)} = \sum_{j_y=0}^{N_y-1} f_{j_x, j_y, j_z} \exp(-2\pi i k_y j_y / N_y)$$

(ただし  $j_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; j_z=0, 1, \dots, N_z-1$ )

--- (数2)

$$c_{k_x, k_y, j_z}^{(2)} = \sum_{j_x=0}^{N_x-1} c_{j_x, k_y, j_z}^{(1)} \exp(-2\pi i k_x j_x / N_x)$$

(ただし  $k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; j_z=0, 1, \dots, N_z-1$ )

--- (数3)

$$c_{k_x, k_y, k_z} = \sum_{j_z=0}^{N_z-1} c_{k_x, k_y, j_z}^{(2)} \exp(-2\pi i k_z j_z / N_z)$$

(ただし  $k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; k_z=0, 1, \dots, N_z-1$ )

--- (数4)

これらの変換を分散メモリ型並列計算機上で行うには、(数2)が $N_x \times N_z$ 組の独立な変換、(数3)が $N_y \times N_z$ 組の独立な変換、(数4)が $N_x \times N_y$ 組の独立な変換からなることを利用すればよい。この考え方に基づく計算方式が、「従来の技術」の項で説明した転置アルゴリズムである。転置アルゴリズムによる計算のフローチャートを図7に示す。この計算は、主に次の4つの処理からなる。

#### <Y方向の変換>

入力装置から3次元のデータ  $f_{j_x, j_y, j_z}$  を入力し(図7の処理23)、直方体状に並べ(処理24)、Z軸に垂直な平面でデータをプロセッサ台数と等しい数に分割し、各部分データをそれぞれ1台のプロセッサのメモリに格納する(処理25)。データの分割の仕方には、ブロック分割、サイクリック分割など、いろいろな方式がありうるが(データ分割方式についての詳細はV. Kumar, A. Grama, A. Gupta and G. Karypis: "Introduction to Parallel Computing", The Benjamin/Cummings Publishing Company, 1994参照。)、たとえばブロック分割の場合、プロセッサ  $p$  ( $0 \leq p \leq P-1$ ) はZ座標  $j_z$  が  $(N_z/P) \times p \leq j_z \leq (N_z/P) \times (p+1) - 1$  をみたす  $N_x \times N_y \times N_z/P$  個のデータを持つ。

この状態で、各プロセッサは自分のメモリに格納された部分データに対し、(数2

)によるY方向の変換を行う(処理26)。

#### <転置処理>

次に、データを図3の6のようにY軸に垂直な面でプロセッサ台数と等しい数に分割し、各部分データをそれぞれ1台のプロセッサのメモリに格納する(処理27)。この場合もデータ分割の仕方はいろいろな方式がありうるが、たとえばブロック分割の場合、プロセッサ $p$  ( $0 \leq p \leq P-1$ ) はY座標 $k_y$ が $(N_y/P) \times p \leq k_y \leq (N_y/P) \times (p+1)-1$ をみたす $N_x \times N_y/P \times N_z$ 個のデータを持つ。このように分割方式を変更するため、プロセッサ $p$ はY方向の変換時に自分が持っていたデータのうち、Y座標 $k_y$ が $(N_y/P) \times p' \leq k_y \leq (N_y/P) \times (p'+1)-1$ をみたす $N_x \times N_y/P \times N_z/P$ 個のデータを、プロセッサ $p'$ に転送する。この処理を転置処理と呼ぶ。

#### <X方向の変換>

各プロセッサは自分のメモリに格納された部分データに対し、(3)式によるX方向の変換を行う(処理28)。

#### <Z方向の変換>

各プロセッサは自分のメモリに格納された部分データに対し、(4)式によるZ方向の変換を行う(処理29)。変換結果のデータを出力装置へ出力する(処理30)。

### 【0008】

以上が従来の転置アルゴリズムによる3次元高速フーリエ変換である。しかしこの計算方式では、「従来の技術」の項で説明したように、途中の転置処理27において全プロセッサ対全プロセッサのデータ転送が必要となり、この部分が性能上のネックになる。

#### (3) 本発明の方法による3次元並列高速フーリエ変換

そこで本発明では、上記の転置アルゴリズムにおけるY方向とX方向の変換をそれぞれ2つの部分処理に分割して演算の順序を変更し、通信と演算のオーバーラップを可能にする。本発明による計算のフローチャートを図1に示す。ただし、本発明による計算は、図7に示した転置アルゴリズムによる計算と、処理25以前の部分、および処理29以後の部分は同じであるので、図1のフローチャートでは処理25より後で、かつ処理29より前の部分のみを示してある。この部分は、次の5つの処理からなる。



### ＜Y方向の変換処理(1)＞

第7図の分割ステップ25で複数プロセッサのメモリに分散格納された部分データのうち、X座標 $j_x$ が偶数の要素のみに対し、(2)式によるY方向の変換を行う(図1の処理33)。部分データの分割について再度述べると、変換対象データを直方体に配列した複素データ列についてZ軸と垂直な面でプロセッサ台数と同じ数に分割する。したがって、上記複素データ列のX軸方向及びY軸方向は分割されていないので、プロセッサ間のデータ転送なしにX方向及びY方向での変換処理を完了できる分散配置となる。しかし、Y方向の変換処理(1)では、このうちX座標 $j_x$ が偶数の要素に対するY方向の変換のみを各プロセッサにて行う。

### ＜Y方向の変換処理(2)＞

各プロセッサは、自分のメモリに格納された部分データのうち、X座標 $j_x$ が奇数の要素のみに対し、(2)式によるY方向の変換を行う。

これと並行して、X座標 $j_x$ が偶数の要素に対しては、転置処理を行う。転置処理においてプロセッサ $p$ は、従来の方法での転置処理と同様、自分が持っているデータのうち、Y座標 $k_y$ が $(N_y/P) \times p' \leq k_y \leq (N_y/P) \times (p'+1) - 1$ をみたす $N_x/2 \times N_y/P \times N_z$ 個のデータをプロセッサ $p'$ に転送する(図1の処理34)。つまりこの転置処理の結果、X座標 $j_x$ のが偶数の要素についてのみ、プロセッサ間で再配置される。その結果、プロセッサ間のデータ転送なしに各プロセッサでZ方向の変換が完了できる状態となる。またX方向については、隣接する要素同士の演算であるX方向の変換の最終ステップのみを除いた $\log_2 N_x 1$ ステップを各プロセッサで実行できる状態となる。

### ＜X方向の変換処理(1)＞

各プロセッサは、自分のメモリに格納された部分データのうち、X座標 $j_x$ が偶数の要素のみに対し、X方向の変換の最終ステップのみを除いた最初の $\log_2 N_x 1$ ステップを行う。このときの計算式は、(数3)の代わりに

$$c_{kx, ky, jz}^{(2')} = \sum_{j_x'=0}^{N_x/2-1} c_{j_x', ky, jz}^{(1)} \exp(-2\pi i k_x 2j_x' / N_x)$$

(ただし $k_x=0, 1, \dots, N_x-1$ ;  $k_y=0, 1, \dots, N_y-1$ ;  $j_z=0, 1, \dots, N_z-1$ )

--- (数5)

とする。

これと並行して、X座標 $j_x$ が奇数の要素に対しては、転置処理を行う。転置処理においてプロセッサ $p$ は、従来の方法での転置処理と同様、自分が持っているデータのうち、Y座標 $k_y$ が $(N_y/P) \times p' \leq k_y \leq (N_y/P) \times (p'+1) - 1$ をみたす $N_x/2 \times N_y/P \times N_z$ 個のデータをプロセッサ $p'$ に転送する（図1の処理35）。これにより、X座標 $j_x$ のが奇数の要素についても、各プロセッサでZ方向の変換が完了でき、またX方向については、隣接する要素同士の演算であるX方向の変換の最終ステップのみを除いた $\log_2 N_x 1$ ステップを各プロセッサで実行できる状態に再配置される。

### 【0009】

#### <X方向の変換処理(2)>

各プロセッサは、自分のメモリに格納された部分データのうち、X座標 $j_x$ が奇数の要素のみに対し、X方向の変換の最終ステップのみを除いた最初の $\log_2 N_x 1$ ステップを行う。このときの計算式は、(数3)の代わりに

$$c_{kx, ky, jz}^{(2'')} = \sum_{j_x'=0}^{N_x/2-1} c_{2j_x'+1, ky, jz}^{(1)} \exp(-2\pi i k_x (2j_x'+1)/N_x)$$

(ただし $k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; j_z=0, 1, \dots, N_z-1$ )

--- (数6)

とする（図1の処理36）。

#### <X方向の変換処理(3)>

各プロセッサは、自分のメモリに格納されたデータを用いて、X方向の変換の最後の1ステップを行う。このときの計算式は、(数5)で計算した $c_{kx, ky, jz}^{(2')}$ 、および(6)式で計算した $c_{kx, ky, jz}^{(2'')}$ を用いて、

$$c_{kx, ky, jz}^{(2)} = c_{kx, ky, jz}^{(2')} + c_{kx, ky, jz}^{(2'')}$$

(ただし $k_x=0, 1, \dots, N_x-1; k_y=0, 1, \dots, N_y-1; j_z=0, 1, \dots, N_z-1$ )

--- (数7)

とする（図1の処理37）。

以上の計算方式では、各プロセッサはY方向の変換処理(2)においてX座標が奇数の要素に対してY方向の変換処理を行うと同時に、X座標が偶数の要素に対してデータの転送処理を行うことができる。また、各プロセッサはX方向の変換処理(1)において、X座標が偶数の要素に対してX方向の変換処理を行うと同時に、X座標が奇数の要素に対してデータの転送処理を行うことができる。これにより、転送のための時間の一部または全部を演算時間に隠蔽することが可能となり、計算効率を向上させることができる。

#### (4) 並列高速フーリエ変換ライブラリ

本発明を並列計算機上で3次元高速フーリエ変換を行うライブラリに適用した例を示す。本ライブラリは、サブルーチン名称をFFT3Dとし、実行するには

CALL FFT3D(N<sub>x</sub>, N<sub>y</sub>, N<sub>z</sub>, P, F, TB, IOPT, IER)

のようにしてすべてのプロセッサで同時にコールする。ここで、N<sub>x</sub>, N<sub>y</sub>, N<sub>z</sub>はフーリエ変換を行うべき3次元データのそれぞれX、Y、Z方向の個数、Pはプロセッサ台数、Fは入力時はフーリエ変換すべき3次元データ{f<sub>j<sub>x</sub>, j<sub>y</sub>, j<sub>z</sub></sub>}、出力時はフーリエ変換結果{c<sub>k<sub>x</sub>, k<sub>y</sub>, k<sub>z</sub></sub>}を格納する配列、TBは変換に用いる複素指数関数の値を格納するテーブル、IOPTはサブルーチンの機能を指定する入力、IERは実行時エラーが生じたか否かを示す出力である。ここで、配列Fは各プロセッサがそれぞれ持つ部分配列であり、Z座標に対してブロック分割を行うので、第p番目のプロセッサp (0 ≤ p ≤ P-1) はZ座標j<sub>z</sub>が(N<sub>z</sub>/P) × p ≤ j<sub>z</sub> ≤ (N<sub>z</sub>/P) × (p+1) - 1をみたすN<sub>x</sub> × N<sub>y</sub> × N<sub>z</sub>/P個のデータのみを持つ。すなわち、第p番目のプロセッサの配列Fには、

$$F(j_x, j_y, j_z') = f_{j_x, j_y, (N_z/P) * p + j_z'}$$

(ただし j<sub>x</sub>=0, 1, …, N<sub>x</sub>-1; j<sub>y</sub>=0, 1, …, N<sub>y</sub>-1; j<sub>z</sub>'=0, 1, …, N<sub>z</sub>/P-1)

--- (数8)

を格納する。したがって各プロセッサの持つ配列Fの大きさはN<sub>x</sub> × N<sub>y</sub> × N<sub>z</sub>/Pである。また、TBは、第1回目のコールで計算した複素指数関数の値を格納しておくテーブルであり、2回目のコールからはここに格納した値を再利用することによ

り、新たな計算が不要となる。また、第1回目のコールではIOPT=1を指定し、このときは複素指数関数のテーブルを作成する。IOPT=2は2回目以降のコールを意味し、このときは既にTBに格納されている値を用いる。

### 【0010】

本ライブラリのフローチャートを図8に示す。ライブラリは、コールされるとまず入力データをチェックし、 $N_x$ ,  $N_y$ ,  $N_z$ とPとが1以上の整数であるかどうか、IOPTが1または2の値であるかどうかなど、入力の有効性を調べる(処理40)。入力データに無効な値が入っていた場合は、IER=1000と設定して(処理41)リターンする。次に、ライブラリが引数で指定した通りにP台のプロセッサでコールされているかどうかをチェックする(処理43)。この条件が満たされていない場合は、IER=2000と設定して(処理44)リターンする。次にIOPTの値をチェックし(処理45)、IOPT=1の場合は各方向のフーリエ変換で用いる複素指数関数の値を前もって計算し、配列TBに格納する(処理46)。

### 【0011】

次に、Y方向およびX方向の変換を行う(処理47)。この変換は、本実施例における「(3) 本発明の方法による3次元並列高速フーリエ変換」の項で述べた方法に従い、Y方向の変換処理(1)(図1の処理33)からX方向の変換処理(3)(図1の処理37)の5つを順に行うことによって終了する。次に、本実施例における「(2) 従来の方法による3次元並列高速フーリエ変換」の項で述べた方法に従い、Z方向の変換を行う(図8の処理48)。これにより、3次元高速フーリエ変換の処理が終了する。終了時には、従来の方法による3次元並列高速フーリエ変換と同様、配列FにはY座標に対してブロック分割されたデータが格納される。すなわち、第p番目のプロセッサpの配列Fには、

$$F(k_x, k_y', k_z) = c_{kx, (N_y/P)*p+k_y', k_z}$$

(ただし  $k_x=0, 1, \dots, N_x-1$ ;  $k_y'=0, 1, \dots, N_y/P-1$ ;  $k_z=0, 1, \dots, N_z-1$ )

--- (数9)

が格納される(図8の処理49)。

## (5) シミュレーションプログラム

本実施例において実行すべき気象計算のための並列プログラムを図9に示す。ここでは、サイズ $N_x \times N_y \times N_z$ の3次元メッシュで計算を行う場合を例にとって説明する。

## 【0012】

本プログラムでは、まず計算領域のサイズ $N_x$ ,  $N_y$ ,  $N_z$ 、温度・風速・圧力などの初期条件、空気の熱伝導率をはじめとする物質定数などのパラメータを入力し（処理51）、計算に必要な前処理を行う。ここで前処理とは、観測によって得られた温度・風速・圧力などのデータに対して補間を行い、計算に必要なメッシュポイントでのデータを得ることである。次に、温度・風速・圧力などのデータを並列計算機の各プロセッサに分割する（処理52）。分割は、前項で述べた3次元高速フーリエ変換ライブラリFFT3Dを利用できるように、Z方向をブロック分割する形で行う。

## 【0013】

これらの処理が終わった後、ループにより各時間ステップでの温度・風速・圧力などの量を順々に求めていく。基礎となる方程式は、風速に対する運動方程式

$$du/dt = -2\Omega \times u - (1/\rho) \nabla p + F_u \quad \text{--- (数10)}$$

質量保存の式

$$d\rho/dt = -\rho \nabla \cdot u \quad \text{--- (数11)}$$

温度変化を表す式

$$dT/dt = -\kappa \nabla^2 T + u \cdot \nabla T \quad \text{--- (数12)}$$

の3本である。ここで、 $u$ は風速、 $p$ は圧力、 $T$ は温度を表し、 $\Omega$ はコリオリ力と呼ばれる地球の自転による力、 $F_u$ はそれ以外の外力、 $\rho$ は空気の密度、 $\kappa$ は空気の熱伝導率を表す。これらの式から次の時刻でのデータの値を求めるには、まずフーリエ変換により格子点上の温度 $T$ 、圧力 $p$ および風速 $u$ を波数空間でのデータに変換し（処理53）、波数空間でこれらのデータを微分し（処理54）、再び波数空間でのデータを逆フーリエ変換して格子点上での温度勾配 $\nabla T$ 、2次微分 $\nabla^2 T$ 、圧力勾配 $\nabla p$ 、速度の発散 $\nabla \cdot u$ を求める（処理55）。この後、これらの量を(8)-(10)式の右辺に代入し、次の時間ステップでの温度・風速・圧力を求め

る(処理56)。なお、フーリエ変換により格子点上のデータを波数空間上のデータに変換して処理を行うのは、その方が微分が精度良く計算できるからであり、本プログラムではこの部分で2次元フーリエ変換ライブラリFFT3Dを用いる。

上記のループでは、各時間ステップ毎に、求める時刻までの計算が終了したかどうかを判定し(処理57)、終了したら、後処理を行い(処理58)、結果を出力する(処理59)。後処理では主に、計算を行うメッシュポイントと結果データが必要な点とがずれている場合に、結果を補間して必要な点での値を計算するなどの処理を行う。

上記の例では気象予測計算を行う場合を例にとって説明したが、本発明の手法は、これ以外の応用例についても、並列計算機上で3次元高速フーリエ変換を用いてシミュレーションを行う場合に適用できることは明らかである。また、上記の例ではフーリエ変換前には3次元データをZ方向にブロック分割し、変換後にはデータはY方向にブロック分割されているとしたが、ブロック分割の代わりに、サイクリック分割やブロックサイクリック分割を用いた場合でも、本発明の手法が適用できることは明らかである。更に、上記の例ではY方向、X方向、Z方向の変換をこの順に行うとしたが、X、Y、Zなどは便宜的に定めた座標軸の名前に過ぎないため、上記の計算方法において $X \rightarrow Y$ 、 $Y \rightarrow Z$ 、 $Z \rightarrow X$ などの名前の付け替えを行った計算方法も、本発明の手法と同じものであることは明らかである。

## 《実施例2》

上記実施例1では3次元フーリエ変換のための計算方法を説明したが、本方式は1次元フーリエ変換の場合へも応用できる。N点のデータに対して1次元フーリエ変換を行うには、 $N_x \times N_y \times N_z = N$ を満たす任意の整数 $N_x$ 、 $N_y$ 、 $N_z$ を定めてデータを $N_x \times N_y \times N_z$ の直方体状に並べ、これに対して「ひねり係数の乗算」という処理を加えた3次元フーリエ変換を行えば良いことが知られている。ここで、ひねり係数の乗算とは、(2)式によってY方向の変換を行った後、中間結果の $c_{j_x, k_y, j_z}^{(1)}$ に対して

$$c_{j_x, k_y, j_z}^{(1)} := c_{j_x, k_y, j_z}^{(1)} \times \exp(2\pi i k_y j_x / (N_x N_y))$$

(ただし $j_x=0, 1, \dots, N_x-1$ ,  $k_y=0, 1, \dots, N_y-1$ ,  $j_z=0, 1, \dots, N_z-1$ )

--- (数 1 3)

という処理を行い、かつ (数 3) によって X 方向の変換を行った後に中間結果の  $c_{kx, ky, jz}^{(2)}$  に対して

< X 方向の変換 >

$$c_{kx, ky, jz}^{(2)} = c_{kx, ky, jz}^{(2)} \times \exp(2\pi i (N_y k_x + k_y) j_x / (N_x N_y N_z))$$

--- (数 1 4)

という処理を行うことである。このひねり係数の乗算は、配列  $c_{jx, ky, jz}^{(1)}$  または  $c_{kx, ky, jz}^{(2)}$  に対する要素毎に独立の処理であるので、実施例 1 の方式においても、Y 方向の変換後と X 方向の変換後にそれぞれ組み入れることができる。これにより、本発明の方式を 1 次元高速フーリエ変換に対しても適用することが可能となる。

《実施例 3》

本発明は更に、2 次元の高速フーリエ変換に対しても適用できる。この場合は、実施例 1 の方式による 3 次元高速フーリエ変換において、単に  $N_z=1$  とすればよい。

#### 【0014】

補足すると、変換対象のデータは X 軸と Y 軸の 2 次元に配列された複素データアレイである。実施例 1 で説明した「Y 方向の変換処理(1)」の前のデータ分割の段階では、上記複素データアレイを X 軸に垂直な直線でプロセッサ台数と等しい数の部分データに分割し、各プロセッサのメモリに分散配置することになる。つまり、プロセッサ間のデータ転送なしに Y 方向の変換が完了できる状態である。しかし「Y 方向の変換処理(1)」では、X 座標  $j_x$  が偶数の要素に対する Y 方向の変換のみを各プロセッサにて行う。次に、Y 方向の変換処理(2)で X 座標  $j_x$  が奇数の要素に対する Y 方向の変換を各プロセッサにて行うのと並行して、X 座標  $j_x$  が偶数の要素はプロセッサ間で再配分を行う。この再配分とは、X 座標  $j_x$  が偶数の要素を Y 軸と垂直な直線でプロセッサ台数と等しい数の部分データに分割して各プロセッサに分散配置する処理であり、この結果、各プロセッサでは X 方向の変換

の最終ステップを除いた $\log_2 N_X 1$ ステップが実行可能となる。さらにX方向の変換処理(1)では、再配分されたX座標 $j_X$ が偶数の要素に対するX方向の変換 ( $\log_2 N_X 1$ ステップ) を各プロセッサにて行うのと並行して、X座標 $j_X$ が奇数の要素のプロセッサ間での再配分を行う。この再配分も上記と同様である。更に、X方向の変換処理(2)、 X方向の変換処理(3)と進むのは実施例 1 と全く同様である。変換対象のデータが2次元データであるので、X方向の変換処理(3)で処理が完了する。

#### 《実施例 4》

本発明による高速フーリエ変換を用いてシミュレーションを行う他の例として、半導体デバイス等における電子構造計算を説明する。電子構造計算では、3次元のメッシュで定義された電子の波動関数 $u(r)$ を、シュレディンガー方程式

$$\nabla^2 u(r) + (E - V(r))u(r) = 0 \quad \text{--- (数 15)}$$

に従って計算することにより、半導体の性質を決定するバンドギャップの大きさや、結晶の構造安定性などを求める。ただし、上式で $h$ はプランク定数、 $m$ は電子の質量、 $E$ は対象とする波動関数のエネルギーレベル、 $V$ は結晶中の原子や他の電子によるポテンシャルエネルギーを表す。

#### 【0015】

(15)式の計算では、波動関数の2次微分 $\nabla^2 u(r)$ が必要であるが、気象計算の例において述べたのと同様な理由により、この部分は $u(r)$ をフーリエ変換により波数空間に移してから計算し、結果を逆フーリエ変換で再び実空間に戻す。したがって、並列計算機上で電子構造計算を行う場合には、この部分で本発明の3次元高速フーリエ変換方法が適用できる。

#### 【0016】

##### 【発明の効果】

以上説明したように、本発明によれば、分散メモリ型並列計算機上での高速フーリエ変換において、演算処理とデータ転送処理を並行して行い、後者の時間の一部または全部を前者の時間に隠蔽することが可能となる。したがって本発明の方法では、従来法に比較して並列化効率を向上させることができる。なお、この向上効果は、本発明を適用する分散メモリ型並列計算機のプロセッサ間通信の性能に依存するが、たとえば16台のプロセッサで $N=256^3$ の問題を処理する場合、20



%～30%程度の実行時間短縮が期待できる。

【図面の簡単な説明】

【図 1】

本発明による並列 3 次元高速フーリエ変換方法における Y 方向及び X 方向の変換の計算方法を示すフローチャート。

【図 2】

従来法による 3 次元高速フーリエ変換方法を示す図。

【図 3】

従来法による並列 3 次元高速フーリエ変換方法を示す図。

【図 4】

X 方向の変換におけるデータの依存関係を示す図。

【図 5】

本発明による並列 3 次元高速フーリエ変換方法における Y 方向及び X 方向の変換の計算方法を示す図。

【図 6】

本発明を適用すべき分散メモリ型並列計算機の構成を示す図。

【図 7】

従来法による並列 3 次元高速フーリエ変換方法を示すフローチャート。

【図 8】

本発明による並列 3 次元高速フーリエ変換ライブラリのフローチャート。

【図 9】

本発明を利用して分散メモリ型並列計算機で気象計算を行う場合のフローチャート。

【符号の説明】

1：NX×NY×NZの直方体状に並べたデータ、2：Y軸に平行なNY個のデータ、3：X軸に平行なNX個のデータ、4：Z軸に平行なNZ個のデータ、5：Z軸に垂直な平面で分割した部分データ、6：Y軸に垂直な平面で分割した部分データ、7：X方向の変換における各ステップでのデータの 1 要素、8：右側の要素の値を計算するために左側の要素の値を使うことを示す線、9：Y方向の変換を行っている要素、10

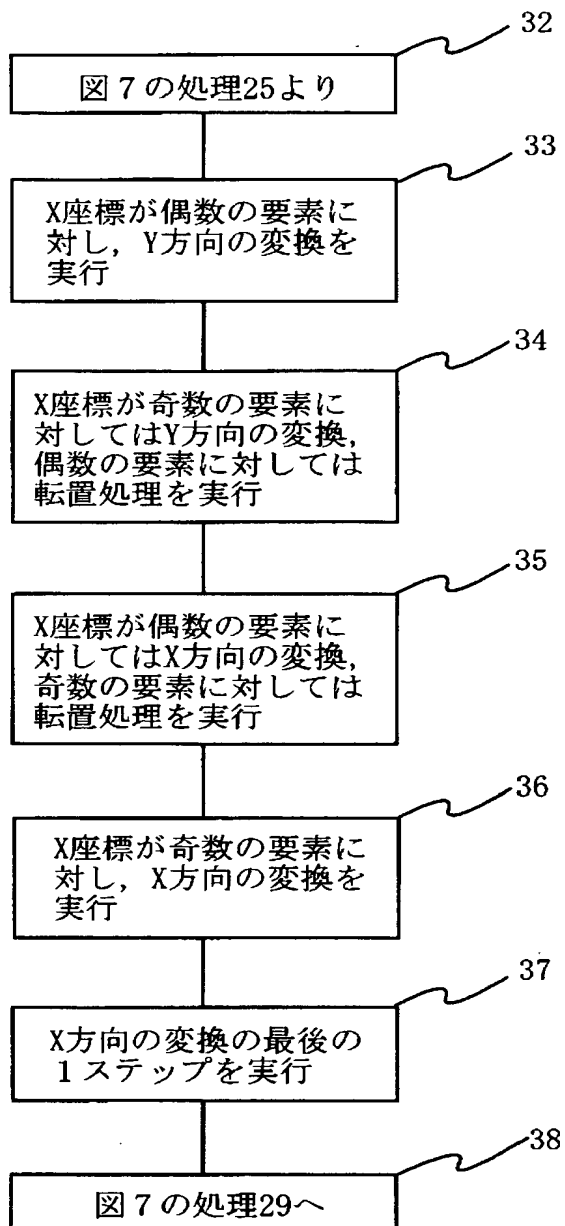
：X方向の変換を行っている要素、11：データ転送を行っている要素、12：何も行わない要素、13：入力装置、14：メモリ、15：プロセッサ、16：ネットワーク、17：処理装置、18：出力装置、19：プログラム、20：データ、21：外部記憶装置、22：スタート、23：変換すべきデータの入力、24：入力データを直方体状に並べる、25：データをZ軸に垂直な面でプロセッサ間に分割、26：Y方向の変換、27：データをY軸に垂直な面で再分割、28：X方向の変換、29：Z方向の変換、30：変換結果の出力、31：終了、32：図7の処理25より継続、33：Y方向の変換処理(1)、34：Y方向の変換処理(2)、35：X方向の変換処理(1)、36：X方向の変換処理(2)、37：X方向の変換処理(3)、38：図7の処理29へ継続、39：ライブラリコール、40：入力データの有効性チェック、41：エラーコードの設定、42：他のプロセッサにコールを通知、43：P台のプロセッサでコールしていることをチェック、44：エラーコードの設定、45：IOPTの値のチェック、46：複素指数関数テーブルの設定、47：Y方向及びX方向の変換、48：Z方向の変換、49：変換結果の格納、50：リターン、51：計算領域の形状・初期条件・パラメータの入力、52：前処理とデータの分割、53：温度及び風速データを端数空間でのデータに変換、54：端数空間で温度勾配及び圧力データを計算、55：温度勾配及び圧力データを格子点上のデータに変換、56：次の時間ステップでの温度と風速データの計算、57：判定処理、58：物理量の計算、59：結果の出力。

【書類名】

図面

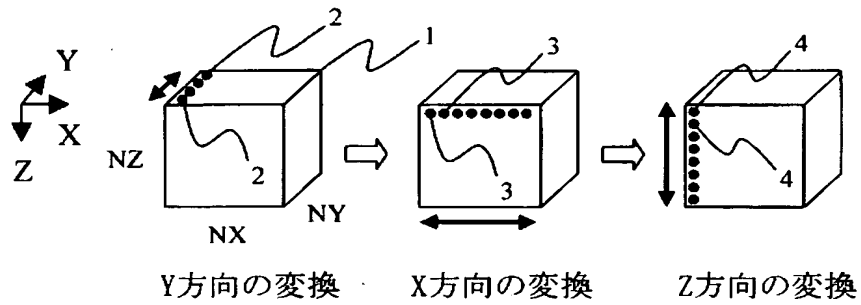
【図 1】

図 1



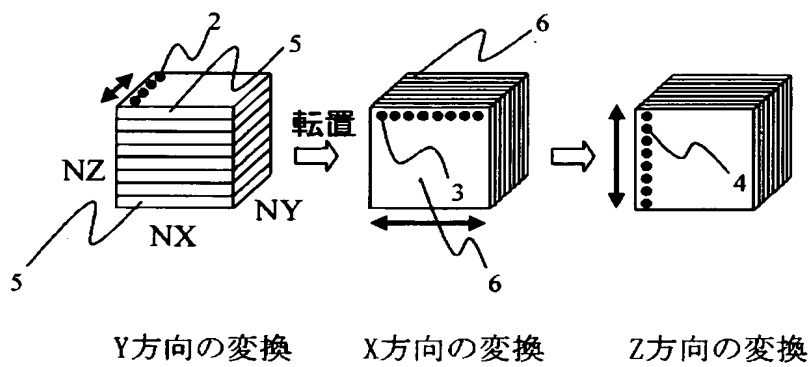
【図 2】

図 2

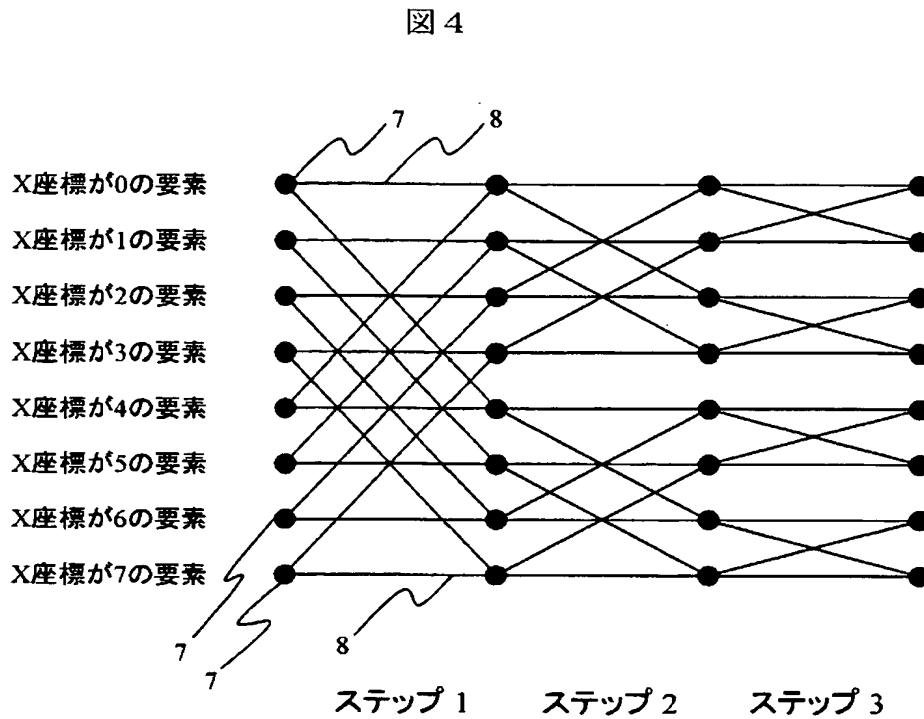


【図 3】

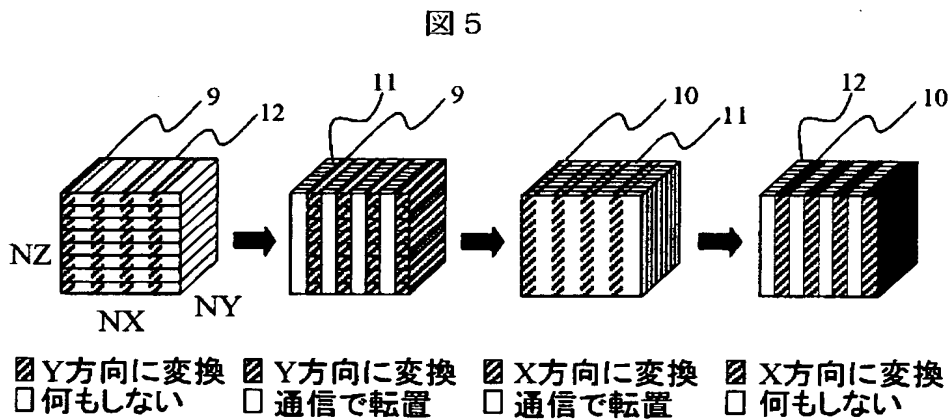
図 3



【図 4】

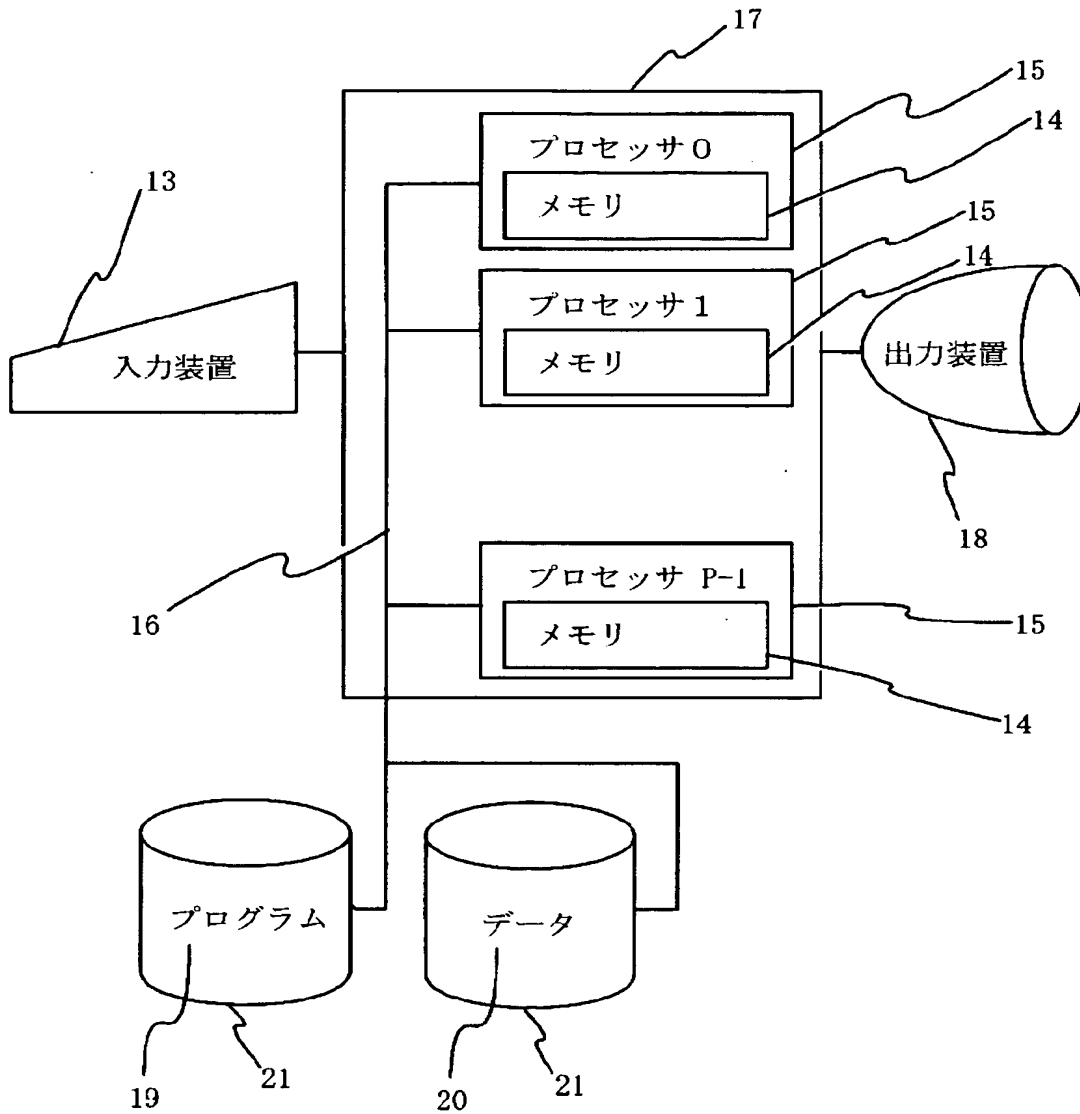


【図 5】



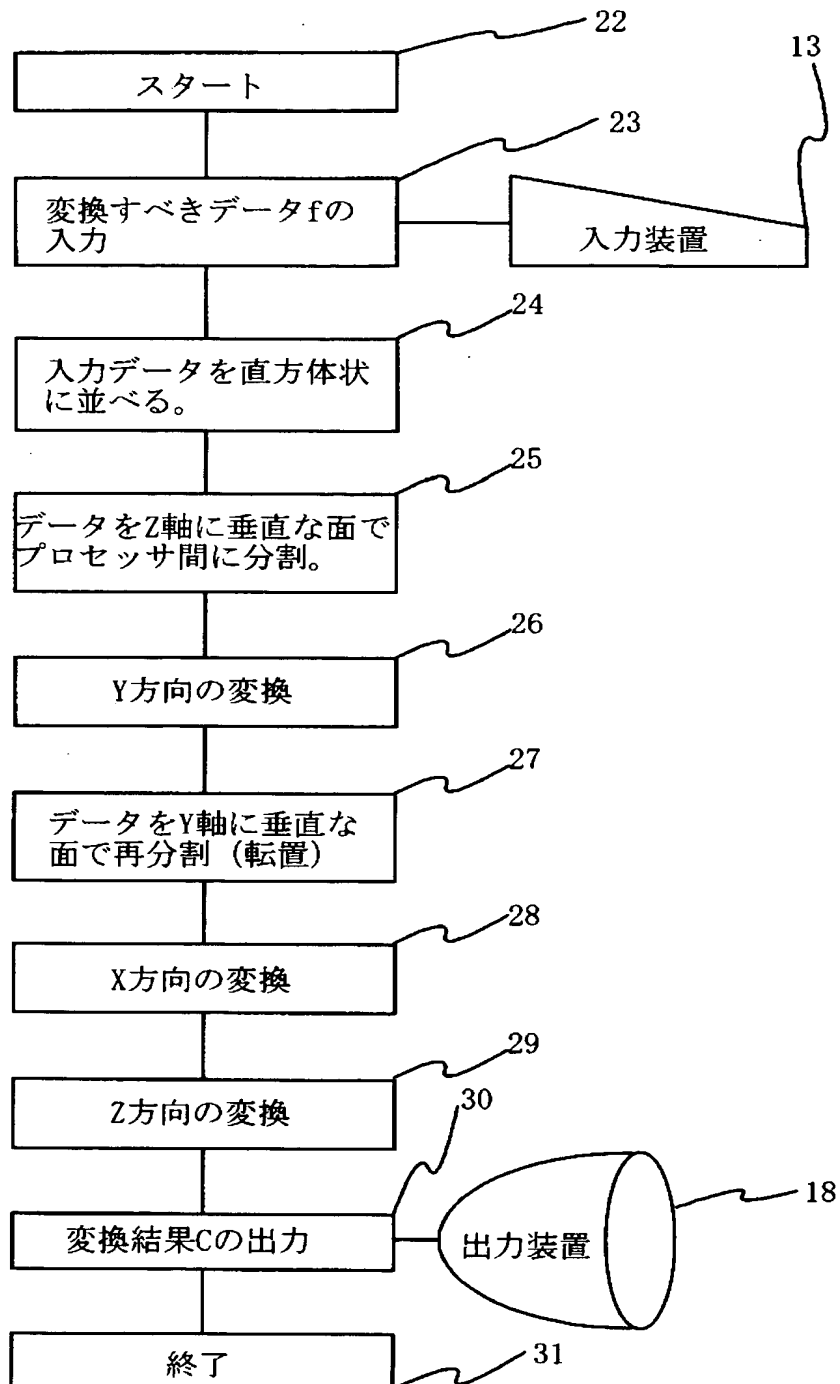
【図 6】

図 6

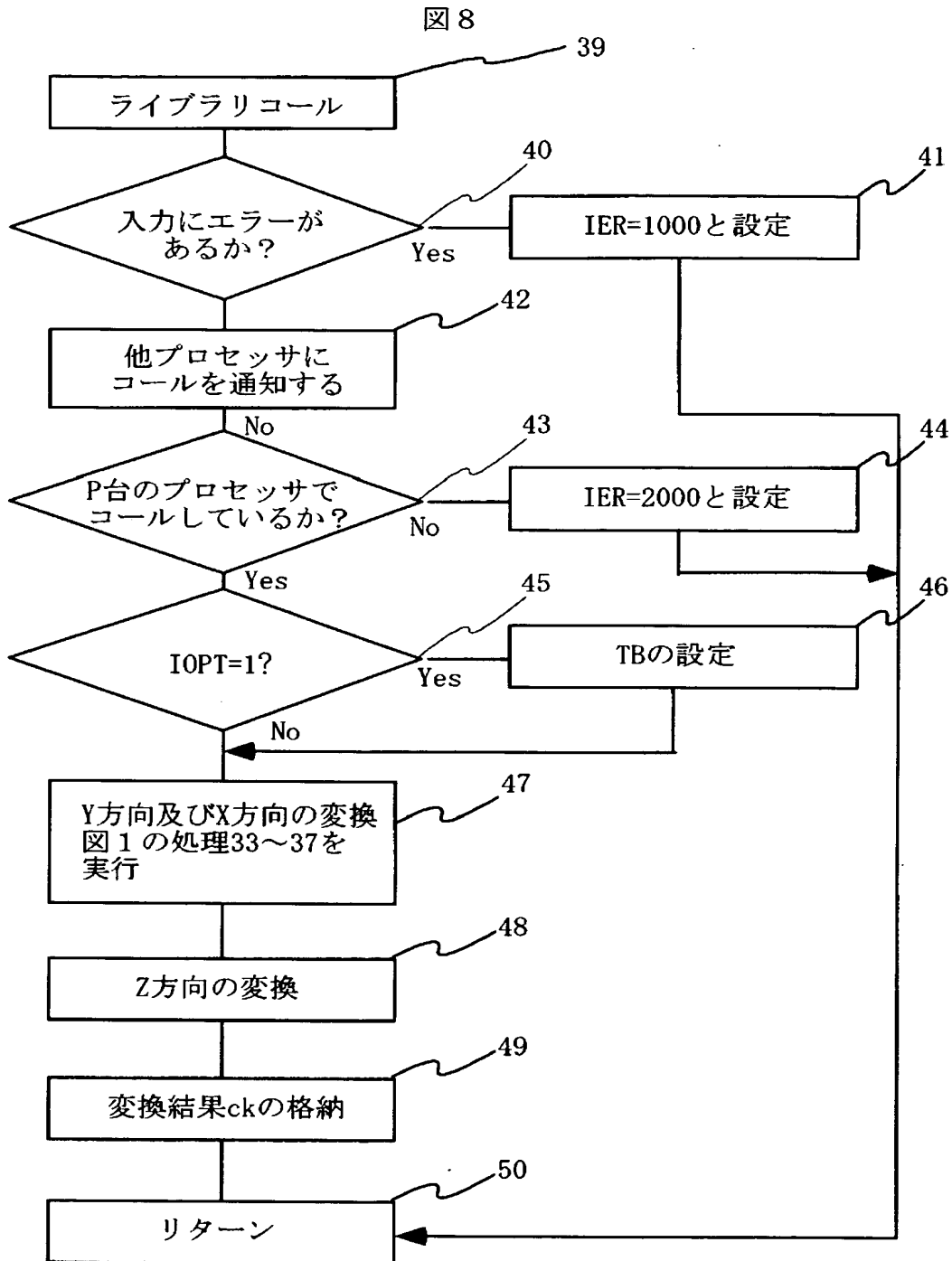


【図 7】

図 7



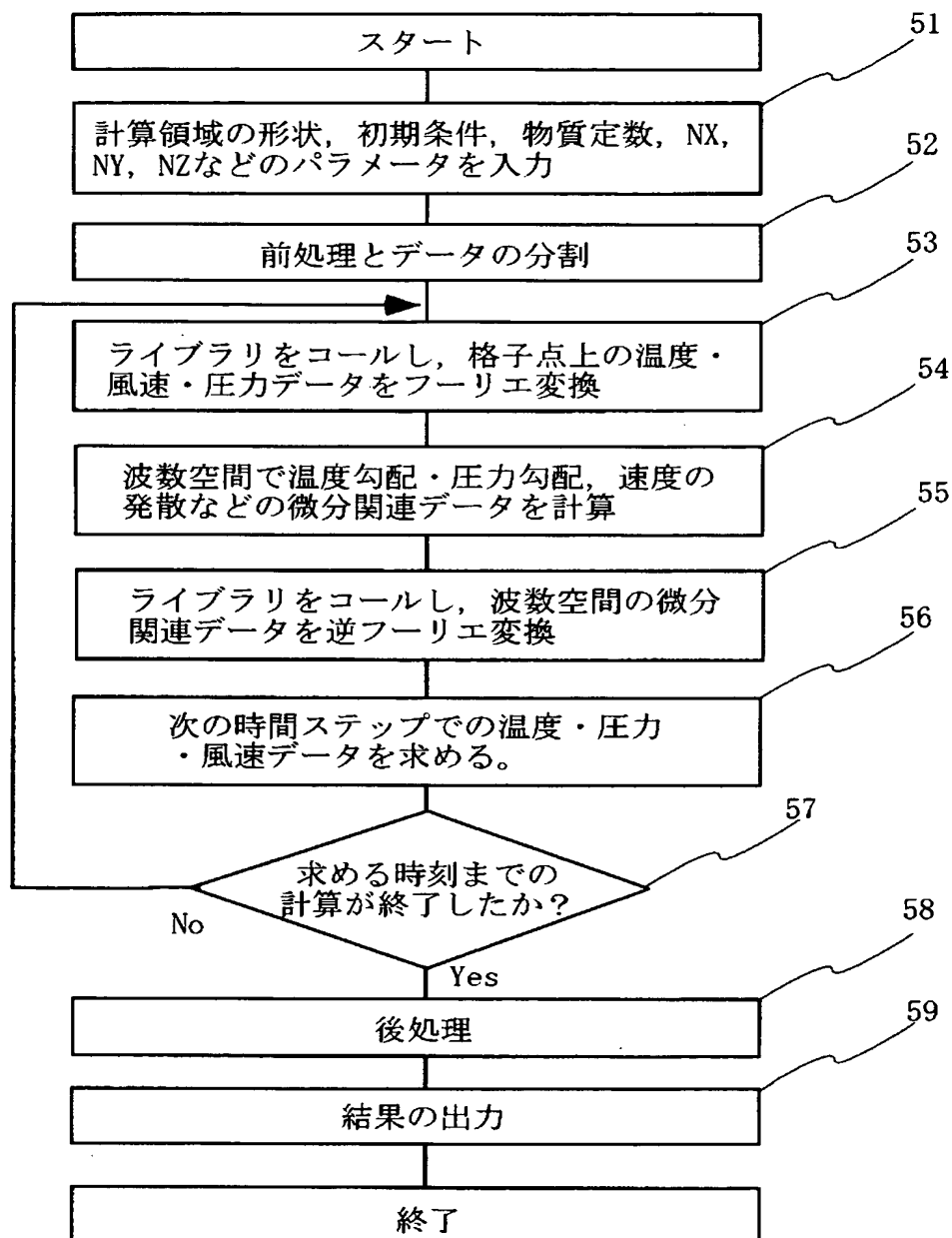
【図 8】





【図 9】

図 9



【書類名】 要約書

【要約】

【課題】 並列計算機による 3 次元の高速フーリエ変換において、プロセッサ間でのデータ転送によるオーバーヘッドを削減し、並列化効率を向上させることを目的とする。

【解決手段】 データを X 座標が偶数の要素と奇数の要素とに分割し、前者に対して Y 方向の変換を行うと同時に後者に対してはプロセッサ間での転置処理を行い（処理 34）、後者に対して X 方向の変換を行うと同時に前者に対してはプロセッサ間での転置処理を行う（処理 35）。

【効果】 プロセッサ間での転置処理のための通信時間を演算時間に隠蔽することが可能となり、従来法に比べて並列化効率を向上させることができる。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 1 4 5 6 0 7
受付番号	5 0 3 0 0 8 5 5 9 5 5
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 5 月 2 6 日

< 認定情報・付加情報 >

【提出日】 平成15年 5月23日

次頁無

特願 2 0 0 3 - 1 4 5 6 0 7

出 願 人 履 歴 情 報

識別番号 [ 0 0 0 0 0 5 1 0 8 ]

1. 変更年月日	1 9 9 0 年 8 月 3 1 日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台 4 丁目 6 番地
氏 名	株式会社日立製作所